

INT J COMPUT COMMUN, ISSN 1841-9836  
Vol.7 (2012), No. 2 (June), pp. 252-263

# Prioritization of Traffic for Resource Constrained Delay Tolerant Networks

G. Fathima, R.S.D. Wahidabanu

## G. Fathima

Adhiyamaan College of Engg  
Hosur. TamilNadu  
fathima\_ace@yahoo.com

## R.S.D. Wahidabanu

Govt. College of Engg  
Salem. TamilNadu  
drwahidabanu@gmail.com

**Abstract:** In networks with common shared wireless medium, the available bandwidth is always valuable and often scarce resource. In addition to it, memory available at nodes (eg., sensor nodes) might be limited relative to the amount of information that needs to be stored locally. As Delay Tolerant Networks (DTNs) rely on node mobility for data dissemination, the high node mobility limits the duration of contact. Besides the issue of contact opportunities between nodes, the bandwidth, available storage at peering nodes and contact duration also affect data forwarding. These factors also influence the mechanisms such as buffer replacement and scheduling policies. So there are secondary problems that routing strategies may need to take care of such as to deal with limited resources like buffer, bandwidth and power. Furthermore, despite inherent delay tolerance of most DTN driving applications, there can be situations where some messages may be more important than the others and expected to get delivered earlier. So considering the network limitations and application requirements, the problem of choosing the messages to be transmitted when a contact opportunity arises and the messages to be dropped when buffer full is formulated. A buffer management policy to address these issues is proposed and analysed in this paper. Additionally the buffer utilization of various DTN routing protocols and the impact of buffer size on the performance of DTN are studied.

**Keywords:** Delay Tolerant Networks, Buffer management, Prioritization of messages, Delivery ratio and Delivery delay.

## 1 Introduction

Communication networks whether they are wired or wireless operate with the assumption of existence of end-to-end path always between source and destination. However, emerging applications such as earth-quake monitoring, habitat monitoring, Vehicular Ad hoc Networks (VANETs) [17] and military ad hoc networks, will likely to change the typical conditions under which networks operate. In fact in such scenarios, networks pose new challenges like frequent disconnections, limited bandwidth, long delays and high bit error rates. The critical fact is that in such situations, the Transmission Control Protocol (TCP) often does not work [26]. To enable those applications to operate under such challenging conditions, a new network paradigm called Delay Tolerant Networking has been proposed by researchers [16]. Networked environments which operate under such intermittent connectivity are referred as Delay/Disruption-Tolerant

Networks (DTNs).

Delay-Tolerant Networking architecture given by [25] is designed to provide support for message based asynchronous communication over network prone to frequent disconnection, long and variable delays. The details of Delay Tolerant Networking architecture are available in [11]. From the literature survey [9], [13], [15], [22], [24] it is understood that a large amount of research has been performed in developing efficient routing algorithms for DTNs. DTNs operate with the principle of store, carry and forward. According to this principle, a node may store a message in its buffer and carry it along for long period of time until it can forward it further. In addition, to achieve high delivery probability, messages are replicated multiple times. Most of the routing protocols in DTN assume that the buffer available as infinite, which is not the case in reality. Therefore the buffers in the node will run out of capacity at certain point of time due to long term storage and extensive message replication. Further, transmission takes place when nodes come into communication range of each other. The node has to decide which of the messages to be transmitted among the messages those are available in the buffer. It is a challenging problem because of the resource constraints like limited bandwidth and limited period of contact due to node mobility. In this paper, a prioritized buffer management approach is proposed which takes care of both: which messages to be transmitted when a new contact arises and which messages to be dropped when buffer is full.

The opportunistic Network Environment (The ONE) simulator [12], [14] which is designed specifically for DTN environment is used for evaluation of the proposed approach. The result of evaluation is compared with other dropping policies. The remainder of this paper is organized as follows: Section 2 gives overview of various DTN routing protocols and their buffer utilization. Additionally, the impact of buffer size on performance of DTN is analysed. The Buffer Management and related work is discussed in section 3. Section 4 discusses about the proposed policy. The simulation setup and the results are discussed in section 5. Section 6 concludes the paper.

## 2 Routing Protocols of DTN and their Buffer Utilization

### 2.1 Overview of Routing Protocols of DTN

Extensive study of different routing mechanism is essential to understand the design of DTNs. The details of various routing protocols of DTN are available in [9], [13], [15]. These protocols differ in the knowledge that they use in making routing decisions and the number of replication they make. Examples of some of the DTN protocols are Direct Delivery, First Contact, Epidemic [4], [5], Spray and Wait [3], [21], PRoPHET [2], and MaxProp [10] routing. Recently, the work in [28] categorized the routing solutions for DTN based on the approach they follow and the information known, as deterministic or scheduled, enforced and opportunistic routing. Deterministic routing approaches are used when knowledge of contact information are known ahead of time. In Enforced routing scheme, special-purpose mobile devices like message ferries proposed in [29] and data mules proposed in [27] are employed. The Opportunistic routing approach relies on arrival of contact opportunities. It is used where there is no knowledge about connectivity or mobility is known a priori and no network infrastructure exists to provide connectivity. In this paper the proposed buffer management policy is used with opportunistic routing approach.

### 2.2 Buffer utilization of DTN Routing Protocols

In this section, the buffer utilization of various routing protocols at different transmission ranges are observed. Buffer utilization is defined as the ratio of total size of the buffer occupied

and the total size of the buffer. It is represented mathematically as

$$\text{Buffer utilization} = \frac{\text{total size of buffer occupied}}{\text{total size of the buffer}}$$

The ONE simulator is used to perform the experimentation. A new simulation environment is created that combines movement modelling, routing simulation, visualization and reporting in a single framework. They are loaded dynamically based on the given configuration files. The Figure 1 depicts the percentage of buffer utilization by various DTN routing protocols. It is observed from the results as shown in Figure 1 that, irrespective of transmission ranges, the buffer utilization is identical and also less for Direct Delivery routing protocol as expected. This is due to the fact that the messages are delivered directly to the destination. The buffer utilization is at the maximum in the transmission range of 250 m for multi-copy routing protocols like Epidemic and Prophet routing as they do replication based on node encounters and they have comparatively more neighbours in this range. Due to the individuality of Spray and Wait routing, it has less buffer usage compared to that of Epidemic and Prophet routing. The utilization of buffer by MaxProp routing though varies with the transmission ranges still, it is noticeably less. It is due to the fact that it uses acknowledgement and removes the stale data from the buffer. It is also evident from the result in Figure 1. In the transmission range of 50 m and below, the possible contact opportunities are less. Therefore the requirement of buffer and their utilization is also less for all routing strategies.

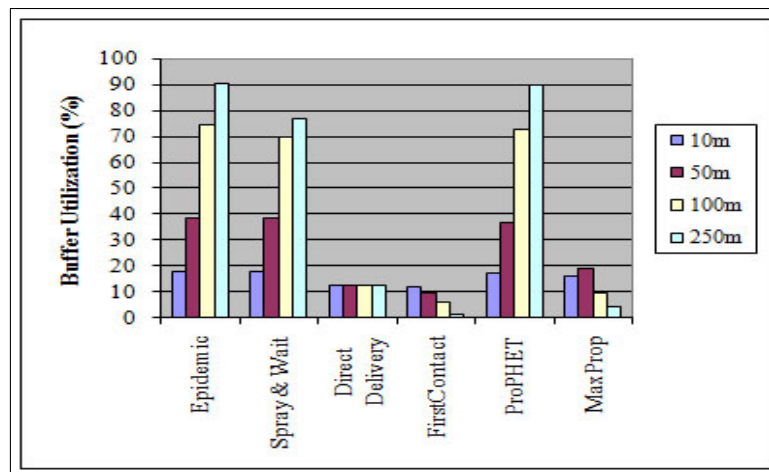


Figure 1: Buffer Utilisation

### 2.3 Impact of Buffer Size

It is essential to understand the impact of buffer size on performance as this resource is limited in reality (example, sensor nodes). Therefore the impact of buffer size is analyzed in this section. It is examined by varying the buffer size in terms of percentage of total number of messages generated. Both Epidemic and Spray and Wait routing are chosen for evaluation. Both the protocols operate on the assumption of infinite buffer and are respectively of type uncontrolled and controlled flooding which requires huge buffer space. The trade-off between the delivery probability and the buffer size is explored at heavy and light traffic load and the results are shown in Figure 2 and 3 respectively. The traffic load is varied by changing the message generation intervals. It is derived from the results that a buffer size of 5-25 % of the generated messages is sufficient to achieve high delivery ratio with reasonable latency.

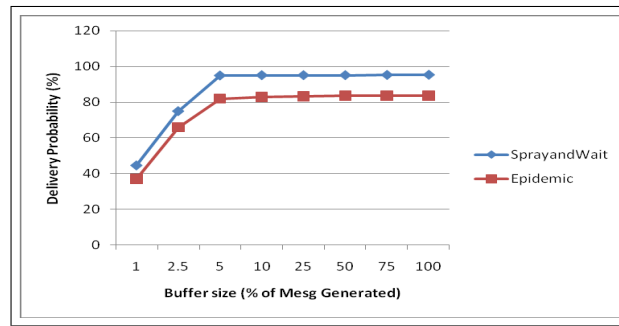


Figure 2: Delivery Probability vs. Buffer size (at Heavy Traffic Load)

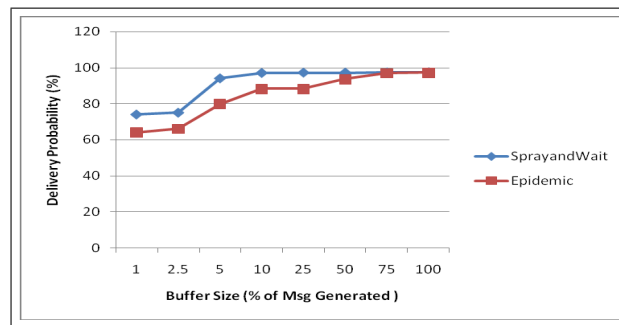


Figure 3: Delivery Probability vs. Buffer size (at Light Traffic Load)

### 3 Buffer Management and Related Work

This section reviews existing literature on buffer management in DTN. The single-copy protocols like Direct Delivery and First Contact routing and multi-copy protocols like Epidemic and Spray and Wait routing transmit messages in FCFS order, that is the messages are transmitted in the order in which they were stored in the buffer. PROPHET routing makes forwarding decision based on delivery predictability of the destination of the message. It needs history of past encounters for calculation of delivery predictability. MaxProp routing [10] assigns priority to messages based on hop count and delivery likelihood. It needs to maintain history of data to estimate delivery likelihood.

RAPID protocol [7] explicitly calculates per-packet utility value for administrator-specified routing metric and forwards the message with highest utility value first. In Prioritized Epidemic Routing [19], transmission and dropping is done based on the priority assigned to each bundle. The priority is based on hop count the bundle has traversed thus far. The Optimal policy in [18], [23] deduces the optimal function for the metrics like delivery ratio and delivery delay independently and the message with smallest utility value is dropped when the buffer is full. The authors in [20] had given a framework for devising optimal routing and scheduling algorithm. It is a centralized algorithm. Irrespective of the routing protocols, several dropping and scheduling policies were proposed in the literature [1]. A different combination of queuing and forwarding strategies had been proposed in [6]. But those policies can be used only with PROPHET routing. It is observed from the study that so far no policies had considered the importance of messages or the requirement of the application.

## 4 Proposed Prioritized Policy

In addition to network and individual node features and capabilities, application specific requirement must be taken into account when developing DTN routing mechanism. More specifically, some applications that use Delay Tolerant Networking service require preferential delivery of certain messages. For example, field agents wish to communicate their findings, regarding environment hazards to other field agents which are more important than the regular findings. However there is no existing solution which can be applied to variety of DTN applications given their requirements. Under such conditions, a forwarding policy is required to serve different types of traffic differently. Consequently it would be necessary to introduce traffic differentiation mechanism and ensure that they get best possible service according to their requirement. Therefore the goal is to determine the policy which maximizes the delivery probability or equivalently minimizes the delivery latency of high priority messages. A buffer management policy to address these issues is proposed and analyzed in detail in this section. The proposed approach attempts to differentiate traffic based on Class-of-Service (CoS) and schedules according to their class. Further the messages are ordered according to their deadline within the class. The assumptions regarding system model is discussed in the following section.

### 4.1 System Model

It is assumed that the network is partially connected with low node density and the node meetings are short lived. When two nodes meet, transmission between them succeeds instantaneously. The messages are stored in the buffer until contact opportunity arises or until storage is full. The bundle protocol specified in [8] is used for transfer of messages in DTN. The Bundle Processing Control Flags Bit in the DTN bundle protocol is used to differentiate the traffic through Class-of-Service (CoS) field. The Class-of-Service is identified in this work by the size of the messages. The priority class is based on the concept proposed by DTN architecture. It is assumed that there are three priority classes of traffic: high, medium and low. The messages are transmitted as a whole from node to node. In this proposed approach, the available buffer is logically divided into three queues to hold the incoming messages. Separate queue is maintained for each priority class as shown in Figure 4. The size of the queue is represented in terms of number of messages. Their sizes are defined at the beginning. They can be varied as the correlation of traffic changes. Initially the buffer is divided equally among all queues.

It is assumed that each node has a buffer  $B$  of size  $b = n(B)$  which is logically divided into three queues:  $B_1$ ,  $B_2$ ,  $B_3$  to accommodate high, medium and low priority bundles respectively such that  $B = B_1 \cup B_2 \cup B_3$ . The size of  $B_1$ ,  $B_2$ ,  $B_3$  is  $b_1$ ,  $b_2$ ,  $b_3$  respectively such that  $b = b_1 + b_2 + b_3$ . A minimum size  $q_{min}$  is specified to reserve space for medium and low priority queues, to avoid the complete negligence of medium and low priority traffic. It is an algorithmic parameter which is set dynamically according to the requirement of the application.

### 4.2 Proposed Approach

The proposed approach comprises (i) *Bundle classifier* which classifies the bundles based on the traffic class as soon as they arrive and stores them in appropriate queue, (ii) *Bundle scheduler* which schedules the bundles based on the priority from high priority to low priority, (iii) *Bundle dropper* which drops the message according to the policy.

Each bundle  $S_i$  in the buffer has a set of information stored with it such as source id, traffic class and Time-To-Live. Several criteria can be used to categorize the bundles such as source id, destination id, size and the TTL of the bundle. In this paper, it is assumed that emergency

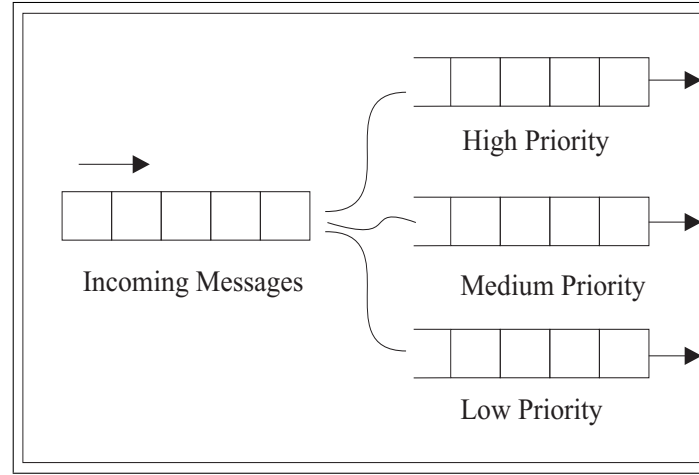


Figure 4: Maintaining Priority Queue

messages are small and have short deadline which is quite a natural phenomenon. With this assumption, the bundles are classified as high priority when the size of the bundle is between 1 and 10KB. The bundles are classified as medium priority when the size of the bundle is between 11KB and 100KB. The bundles are classified as low priority when the size of the bundle is between 101KB and 1000KB. The Bundle Classifier is a function of newly arrived bundle  $S_{new}$  such as

$$f(S_{new}) = \begin{cases} (S_{11}, S_{12}, \dots, S_{1b1}) \cup (S_{new}) & \text{if } S_{new} \text{ is of high Priority} \\ (S_{21}, S_{22}, \dots, S_{2b2}) \cup (S_{new}) & \text{if } S_{new} \text{ is of medium Priority} \\ (S_{31}, S_{32}, \dots, S_{3b3}) \cup (S_{new}) & \text{if } S_{new} \text{ is of low Priority} \end{cases}$$

Where  $(S_{11}, S_{12}, \dots, S_{1b1})$  are the messages in high priority queue,  $(S_{21}, S_{22}, \dots, S_{2b2})$  are the messages in medium priority queue,  $(S_{31}, S_{32}, \dots, S_{3b3})$  are the messages in low priority queue.

The bundle classify procedure is shown in Algorithm 1.

**Algorithm.1** Bundle Classify Procedure

```

Bundle_classify()
Receive bundle;
If buffer full
    Call Bundle_drop procedure;
Else
    Check for Class-of-Service of the bundle;
If received bundle is expedited message,
    Store it in high priority queue;
Else if received bundle is normal message,
    store it in medium priority queue;
Else if received bundle is Bulk message,
    Store it in low priority queue;

```

Bundle scheduler is invoked when contact opportunity arises. The number of messages that can be transmitted is limited by the bandwidth and the duration of the contact between the nodes. In such cases only few messages are transmitted and the order in which the messages

are transmitted is significant. Bundle scheduler transmits the bundles from high priority to low priority. The procedure of bundle scheduler is shown in Algorithm 2. The selection of the bundle to be transmitted  $S_t$  is represented mathematically as follows:

$$S_t = \begin{cases} S_i | S_i \in B_1 & \text{if high priority queue is not empty} \\ S_i | S_i \in B_2 & \text{if medium priority queue is not empty} \\ S_i | S_i \in B_3 & \text{if low priority queue is not empty} \end{cases}$$

The messages whose destination encountered are the first to be transmitted irrespective of the scheduling policy adopted and the same may be deleted from the buffer. Nodes do not delete messages that are forwarded to other nodes (other than destination) as long as there is sufficient space available in the buffer.

**Algorithm.2** Bundle Schedule Procedure

```

Bundle_schedule()
while high priority queue is not empty,
    transmit bundle from high priority queue;
while medium priority queue is not empty,
    transmit bundle from medium priority queue;
while low priority queue is not empty,
    transmit bundle from low priority queue;

```

Once the buffer is full, the Bundle dropper is invoked. As the TTL of the bundle expires, it is dropped automatically. The low and medium priority bundles are dropped to give room for high priority bundles. It is also taken care that a node should not drop its own bundle (source) to give room for newly arrived bundles. The idea of giving priority to source bundles has been proposed in [19], and was shown to improve the average delivery ratio. So the same idea is followed here. The Bundle drop procedure is shown in Algorithm 3. There are three possible classes of bundles that arrive like high priority, medium priority and low priority. Bundle dropping is a function which selects the bundle  $S_d$  to be dropped based on the priority of the bundle that arrive. The identification of the bundle to be dropped is mathematically represented as follows:

Case 1: when high priority bundle arrives

$$S_d = \begin{cases} S_i | S_i \in B_3 & \text{if } L_{min} > q_{min} \\ S_i | S_i \in B_2 & \text{if } M_{min} > q_{min} \\ S_i | S_i \in B_1 & \text{otherwise} \end{cases}$$

Case 2: when medium priority bundle arrives

$$S_d = \begin{cases} S_i | S_i \in B_3 & \text{if } L_{min} > q_{min} \\ S_i | S_i \in B_2 & \text{otherwise} \end{cases}$$

Case 3: when low priority bundle arrives

$$S_d = S_i | S_i \in B_3$$

**Algorithm.3** Bundle Drop Procedure

```

Bundle_drop( )
Let  $L_{min}$  and  $M_{min}$  be the currently occupied space by low priority and medium priority
bundles respectively.
Let  $q_{min}$  be the threshold value which controls the minimum space for low and medium
priority bundles.
Case 1: High priority bundle arrives
  If  $L_{min} > q_{min}$ 
    Drop bundle from low priority and the room is added to high priority queue;
  Else if  $M_{min} > q_{min}$ 
    Drop bundle from medium priority and the room is added to high priority queue;
    Else the remaining time of existing bundles of high priority queue is compared with
    newly arrived bundle. The bundle with least remaining time is dropped;
Case 2: Medium priority bundle arrives
  If  $L_{min} > q_{min}$ 
    Drop bundle from low priority and the room is added to Medium priority queue;
  Else the remaining time of existing bundles of medium priority queue is compared with
  newly arrived bundle. The bundle with least remaining time is dropped;
Case 3: Low priority bundle arrives
  The remaining time of existing bundles of low priority queue is compared with
  newly arrived bundle. The bundle with least remaining time is dropped;

```

By providing differentiated service based on the class, the best effort service has been enhanced. Moreover the proposed policy takes a state less approach that minimizes the need for nodes in the network to remember anything about flows. It is more practical to implement. The messages are marked in a way that describes the service level that they should receive. These markings are used to provide appropriate service without the need to remember extensive state information for every flow. The metrics used to evaluate the performance of DTN are the delivery ratio and average delivery delay. Both the metrics are defined below.

$$\text{Delivery ratio} = \frac{\text{number of messages delivered}}{\text{total number of messages sent by the sender}}$$

Delivery delay = Average(time taken by all the messages to reach from source to destination)

When compared to other approaches of [18], [19], [20], [23], the proposed approach has the credit of no requirement to maintain state information. Moreover it has less overhead than other approaches as there is no exchange of control traffic before bundle exchange. The proposed approach avoids the complete negligence of medium and low priority messages by reserving minimum space for them.

## 5 Simulation Results and Analysis

To evaluate the performance of proposed policy, experimentation is done using The ONE simulator. The simulation environment consists of sparsely distributed mobile nodes. They are capable of communicating when they are in the communication range of one another. The parameter of the nodes like buffer size, transmit range, transmit speed, number of nodes are set as mentioned in the Table 1. Further,  $q_{min}$  is an algorithm parameter which is set as 10% of the messages generated. The environment where nodes move randomly is considered. So mobility model is set as Random Waypoint Model, in which nodes move independently to a randomly chosen destination. Priority based scheduling is not a basic primitive of DTN routing. Therefore



it has to be incorporated with any of the routing algorithms. Epidemic routing is chosen as baseline for evaluation due to its simplicity and recognition as unbeatable routing protocol from the point of reliable delivery. Furthermore it is suitable for opportunistic environment as it does not rely on previous contact information. However the proposed policy can be incorporated with any of the multi-copy routing protocols.

**Table.1** Simulation parameters

Parameters	Values
Number of Nodes	50
Transmit Range ( $m$ )	250
Transmit speed ( $Mbps$ )	2
Node Speed ( $km/hr$ )	10 - 50
TTL of message ( $min$ )	60
Buffer size ( $MB$ )	1500
Message size	10 KB - 1 MB
Number of Messages /min	4 - 20
Movement Model	Random Waypoint Model
Simulation Time ( $min$ )	160

The performance of Epidemic routing under different buffer management policies are compared in terms of metrics like delivery probability and average delivery latency. Figure 5 and 6 depict the behavior of the proposed policy versus other dropping policies like Drop Front (DF), Drop Old (DO) and Drop Random (DR) with respect to delivery probability and average delivery latency respectively at various traffic loads. The traffic load is increased by increasing the number of messages generated per interval. It is observed from the result that as and when the traffic load increases, the delivery probability decreases. It also shows that incorporating the prioritized policy does not degrade the performance when compared to other policies.

From the literatures [6], [15] it is noticed that DF policy results in highest delivery ratio and DO

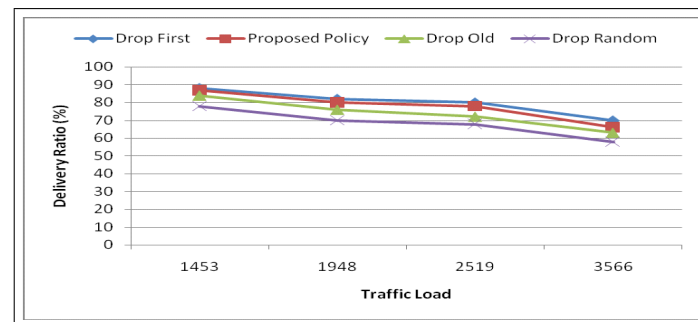


Figure 5: Delivery ratio as a function of Load

results in least average delivery latency. The simulation results shown in Figure 5 and 6 support it. The rationale behind this result is that messages nearing the deadline may get automatically removed from the buffer on the expiry of their deadline. So forcing such messages to drop will not decrease the delivery ratio. Furthermore, high priority messages with earliest deadline are forwarded first. So they have more chances of reaching the destination quickly than other messages before missing their deadline. As the size of high priority message is less compared to other priority class, the number of messages that are transmitted per contact duration is also more. This results in good delivery ratio.

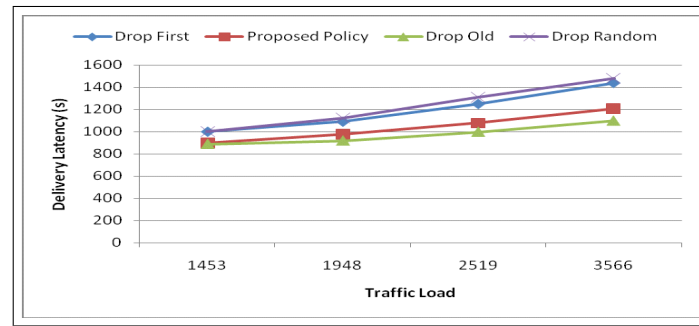


Figure 6: Delivery Latency as a function of load

## 6 Conclusion

The paper studies the buffer utilization of different routing protocols and the impact of buffer size on performance. The work targets on application that requires preferential delivery in opportunistic DTN environment. The prioritized approach presented in this paper differentiates the traffic based on Class-of-Service and does scheduling and dropping based on their priorities. Thereby it ensures the delivery of high priority messages first with least latency satisfying the application requirement. The proposed policy is more suitable and advantageous in strict resource constrained environment with emergency applications. The service required can be specified by the application. So it can be used in vehicular networks where accident notification messages are more important than other messages. In order to avoid starvation of low priority messages, weighted fair queuing can be used which is carried as future work.

## Bibliography

- [1] James. A. Davis, Andrew H. Fagg, and Brian N. Levine, N., Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks, in *Proceedings of International Symposium on Wearable Computing*, pp. 141-148, 2001
- [2] Anders Lindgren, Avri Doria and Olov Schelen, Probabilistic Routing in Intermittently Connected Networks, *Springer LNCS*, Vol. 3126, pp. 239-254, 2004
- [3] Spyropoulos, T., Psounis, K. and Raghavendra, C.S., Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks, in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking*, 2005
- [4] Alan Demers, Dan Greene, Carl Houser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart and Doug Terry, Epidemic Algorithms for Replicated Database Maintenance", in *Proceedings of ACM Symposium on Principles of Distributed Computing*, pp. 1-12, 1987
- [5] Amin Vahdat and David Becker, Epidemic Routing for Partially-Connected Ad Hoc Networks, Technical Report CS-200006, 2000
- [6] Anders Lindgren and Kaustubh Phanse, S., Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks, in *Proceedings of International Conference on Communication System softWARE and MiddlewaRE -COMSWARE*, 2006

- 
- [7] Aruna Balasubramanian, Brian Levine and Arun Venkataramani, DTN Routing As A Resource Allocation Problem, *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 4, 2007
  - [8] Scott, K. and Burleigh, S., Bundle Protocol Specification, RFC 5050, 2007
  - [9] Evan Jones, P.C. and Paul Ward, A.S., Practical Routing in Delay-Tolerant Networks, *IEEE Transaction on Mobile Computing*, 6(8):943-959, 2007
  - [10] Burgess, J., Gallagher, B., Jensen, D., and Levine, B.N., MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks, in *Proceedings of IEEE International Conference on Computer Communications*, pp. 1-11, 2006
  - [11] Fall, K., A Delay-Tolerant Network Architecture for Challenged Internets, in *Proceedings of SIGCOMM'03*, 2003
  - [12] Are Keranen, Jorg Ott and Teemu Karkkainen, The ONE simulator for DTN protocol evaluation, in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pp. 1-10, 2009
  - [13] Sushant Jain, Kevin Fall and Rabin Patra, Routing in Delay Tolerant Networks, *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 4, 2004
  - [14] TKK/COMNET. Project page of the ONE simulator. 2008. [www.netlab.tkk.fi/tutkimus/dtn/theone/](http://www.netlab.tkk.fi/tutkimus/dtn/theone/)
  - [15] Zhensheng Zhang, Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview And Challenges, *IEEE Communication Surveys and Tutorials*, 8(1):24-37, 2006
  - [16] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K. and Weiss, H., Delay Tolerant Networking Architecture, IETF Network working group, RFC 4838, 2007
  - [17] Basu, P. and Little T.D.C., Networked Parking Spaces: Architecture And Applications, in *Vehicular Technology Conference*, Vol.2, pp. 1153-1157, 2002
  - [18] Amir Krifa, Chadi Barakat and Thrasyvoulos Spyropoulos, Optimal Buffer Management Policies for Delay Tolerant Networks, in *Proceedings of IEEE Conference on SECON*, pp. 260-268, 2008
  - [19] Ramnathan, R., Hansen, R., Basu, P., Rosales Hain, R. and Krishnan, R., Prioritized Epidemic Routing For Opportunistic Networks, in *Proceedings of the 1st International MobiSys workshop on Mobile Opportunistic Networking*, pp. 62-66, 2007
  - [20] David Hay and Paola Giaccone, Optimal Routing And Scheduling For Deterministic Delay Tolerant Networks, in *Proceedings of International Conference on Wireless On-Demand Network Systems and Services*, pp. 27-34, 2009
  - [21] Spyropoulos, T., Psounis, K. and Raghavendra, C.S., Efficient Routing in Intermittently Connected Mobile Networks: The Multi-Copy Case, *IEEE/ACM Transactions on Networking*, 16(1):77-90, 2008
  - [22] Elizabeth Daly, M. and Mads Haahr, The Challenges of Disconnected Delay-Tolerant MANETs, *Elsevier Ad Hoc Networks Journal*, 8(2):241-250, 2010

- 
- [23] Amir Krifa, Chadi Barakat, Thrasyvoulos Spyropoulos, An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks, *IEEE WoWMoM*, 2008
  - [24] Delay tolerant networking research group, [Online]. Available: <http://www.dtnrg.org>
  - [25] Fall, K. and Farrell, S., DTN: an Architectural Retrospective, *IEEE Journal on Selected Areas in Communications*, 26(5):828-836, 2008
  - [26] Farrell, S., Cahill, V., Geraghty, D., Humphreys, I. and McDonald, P., When TCP Breaks: Delay- and Disruption- Tolerant Networking, *IEEE Internet Computing*, 10(4):72-78, 2006
  - [27] Shah, R.C., Roy, S., Jain, S. and Brunette, W., Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks, in *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 30-41, 2003
  - [28] Thrasyvoulos Spyropoulos, Rao Naveed Rais, Thierry Turletti, Katia Obraczka and Athanasios Vasilakos, Routing for Disruption Tolerant Networks: Taxonomy And Design, *Wireless Networks*, Vol. 16, No. 8, 2010
  - [29] Zhao, W., Ammar, M. and Zegura, E., A Message Ferrying Approach For Data Delivery In Sparse Mobile Ad Hoc Networks, in *Proceedings of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing, MobiHoc*, pp. 187-198, 2004